

```
-- control_mux
-- version 3 12/10/2002
-- MUX THAT SELECTS BETWEEN THE fram download and the i2c control
library IEEE;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_arith.all;
library synplify;
use synplify.attributes.all;

entity control_mux is port

-- I2C core connections
(
-- to FRAM DOWNLOAD BLOCK
ack_out_FD          : out std_logic;
command_ack_FD      : out std_logic;
I2C_busy_FD         : out std_logic;
fram_Q_FD           : out std_logic_vector(7 downto 0);

-- fram download BIT CONTROLS where the signals come from
FRAM_DOWNLOAD        : in std_logic;

start_bit            : out std_logic;
stop_bit             : out std_logic;
read_bit             : out std_logic;
write_bit            : out std_logic;
ack_in               : out std_logic;
data_wr_slave        : out std_logic_vector(7 downto 0);

command_ack          : in std_logic;
ack_out              : in std_logic;
I2C_busy             : in std_logic;
data_rd_slave        : in std_logic_vector(7 downto 0);

-- enable for I2C output pins
I2C_enable           : out std_logic_vector(6 downto 0);

-- EXTERNAL FRAM CONNECTIONS
fram_address          : out std_logic_vector(12 downto 0);
fram_OE                : out std_logic;
fram_CE                : out std_logic;
fram_WE                : out std_logic;
fram_D                 : out std_logic_vector(7 downto 0);

-- from FRAM DOWNLOAD BLOCK
start_bit_FD          : in std_logic;
stop_bit_FD            : in std_logic;
write_bit_FD           : in std_logic;
data_wr_slave_FD       : in std_logic_vector(7 downto 0);

fram_address_FD        : in std_logic_vector(9 downto 0);
fram_OE_FD              : in std_logic;
fram_CE_FD              : in std_logic;

I2C_enable_FD          : in std_logic_vector(6 downto 0);
-----
-- from/to I2C CONTROL BLOCK
start_bit_IC            : in std_logic;
stop_bit_IC              : in std_logic;
read_bit_IC              : in std_logic;
write_bit_IC              : in std_logic;
ack_in_IC                : in std_logic;
data_wr_slave_IC         : in std_logic_vector(7 downto 0);

ack_out_IC              : out std_logic;
command_ack_IC           : out std_logic;
```

```
I2C_busy_IC      : out std_logic;
data_rd_slave_IC : out std_logic_vector(7 downto 0);

fram_address_IC  : in std_logic_vector(9 downto 0);
fram_OE_IC       : in std_logic;
fram_WE_IC       : in std_logic;
fram_CE_IC       : in std_logic;
fram_D_IC        : in std_logic_vector(7 downto 0);
fram_Q_IC        : out std_logic_vector(7 downto 0);

I2C_enable_IC    : in std_logic_vector(6 downto 0);

-- EXTERNAL FRAM CONNECTIONS
fram_Q           : in std_logic_vector(7 downto 0));
-----  
end control_mux;

architecture rtl of control_mux is
attribute syn_radhardlevel of rtl : architecture is "tmr";
begin
    ack_out_FD      <= ack_out;
    command_ack_FD <= command_ack;
    I2C_busy_FD     <= I2C_busy;
    ack_in          <= ack_in_IC;
    ack_out_IC      <= ack_out;
    command_ack_IC <= command_ack;
    I2C_busy_IC     <= I2C_busy;
    data_rd_slave_IC <= data_rd_slave;
    fram_Q_FD       <= fram_Q;
    fram_Q_IC       <= fram_Q;
    fram_D          <= fram_D_IC;
    read_bit         <= read_bit_IC;
    fram_address(12 downto 10) <= "000";

    start_bit       <= start_bit_FD      when (FRAM_DOWNLOAD = '1') else start_bit_IC;
    stop_bit         <= stop_bit_FD       when (FRAM_DOWNLOAD = '1') else stop_bit_IC;
    write_bit        <= write_bit_FD      when (FRAM_DOWNLOAD = '1') else write_bit_IC;
    data_wr_slave   <= data_wr_slave_FD when (FRAM_DOWNLOAD = '1') else data_wr_slave_
IC;
    fram_address(9 downto 0)      <= fram_address_FD  when (FRAM_DOWNLOAD = '1') else fr
am_address_IC;
    fram_OE          <= fram_OE_FD      when (FRAM_DOWNLOAD = '1') else fram_OE_IC;
    fram_WE          <= '1'           when (FRAM_DOWNLOAD = '1') else fram_WE_IC;
    fram_CE          <= fram_CE_FD      when (FRAM_DOWNLOAD = '1') else fram_CE_IC;
    I2C_enable        <= I2C_enable_FD  when (FRAM_DOWNLOAD = '1') else I2C_enable_IC;

end rtl;
```